

## REMARKS

The present response is intended to be fully responsive to all points raised by the Examiner in the Office Action and is believed to place the application in condition for allowance. Favorable reconsideration and allowance of the application is respectfully requested.

The Applicant expresses appreciation to the Examiner for clarifying the grounds of rejection in the Office Action.

The application as examined includes claims 27 – 38, 40 – 51, and 71 – 78. In the present response, claims 42 and 49 are amended. Claims 27 – 38, 40 – 41, 43 – 48, 50 – 51, and 71 – 78 are unchanged. No additional claims are canceled, and no new claims are added.

### ***35 U.S.C. §112 Second Paragraph Rejections***

Claims 42 – 51 and 73 – 78 stand rejected under 35 U.S.C. §112 as being indefinite, because claims 42 and 49 recite the limitation “said set of non-unique values”, with insufficient antecedent basis for this limitation in the claim. The Applicant is overcoming this rejection by amending claims 42 and 49 to recite “a set of non-unique values”.

### ***35 U.S.C. §103(a) Rejections***

Claims 27 – 38 and 40 – 51 stand rejected under 35 U.S.C. §103(a) as being unpatentable over *Genesys-MP: User’s Guide* (“Genie”) in view of “A Simulation Based Approach to Architectural Verification of Multiprocessor Systems” (“Saha”).

Claims 71 – 78 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Genie in view of Saha in view of “Development and Validation of a Hierarchical Memory Model Incorporating CPU- and Memory- Operation Overlap” (“Luo”).

The present invention provides a computer program product for validating a multi-processor design by simulating program execution for a test program having at least two simulated processes which access mutually-dependent non-adjacent

resources, and by creating one or more tagged value-lists incorporating a set of non-unique values associated with a combination identifier which includes a string of literals identifying a particular outcome of the test program. The contents of the resources are then compared with the non-unique values in the tagged value-lists to validate the processor design for the test program.

Genie describes the use and operation of the “Genesys” multi-processor test system.

Saha describes verifying coherency in weakly ordered shared memory multiprocessor systems through the use of “data coloring”.

The Applicant respectfully traverses the 35 U.S.C. §103(a) rejection based on Genie in view of Saha, on the grounds that Saha fails to disclose or reasonably suggest a set non-unique values as defined in the present description.

The Office Action states:

“As previously stated by Examiner, the term {‘non-unique’} has been interpreted in view of the specification, and as per the specification, *a set of non-unique values is a set of permissible or valid values.*” (Office Action page 2, point 3.i., under “Examiner notes”, emphasis added).

The Applicant respectfully disagrees, in that the present specification does not define or describe a set of non-unique values as “valid” values. The present specification is clear that a set of non-unique values is a set of *permissible* values, which are not the same as “valid” values. To show that *permissible* values are not the same as *valid* values according to the present invention, refer for example to the following (emphasis added):

[Present application, paragraph 0021]: In another aspect of the method the first member of the first set of non-unique values includes a first value-list, and the second member of the second set of non-unique values includes a second value-list, respective *elements of the first value-list being permissible values* of the first resource and adjacent resources thereof, and respective *elements of the second value-list being permissible values* of the second

resource and adjacent resources thereof. Verification is accomplished by verifying an equality between a content of the first resource and adjacent resources thereof with corresponding elements of the first value-list, and verifying an equality between a content of the second resource and adjacent resources thereof with corresponding elements of the second value-list.

The above passage uses the term “permissible”, not the term “valid” to describe a set of non-unique values. Whether or not these values are “valid” is a separate issue according to the present invention, and is disclosed in the present specification, for example as shown in the following (emphasis added):

[Present application , paragraph 0181] As the program of Listing 4 demonstrates, there may be cases where there exist two resources, *each with non-unique results, but where not every combination is valid*. In Listing 4, memory 3000 and register R2 may contain either of the values 1 or 2 and register R3 may contain either of the values 2 or 4. In the embodiment of FIG. 3, all of the eight combinations of these value would be accepted. However, *only the combinations {1,1,2}, {1,2,4}, and {2,2,4}, for the memory 3000 and the registers R2 and R3 respectively, are permissible*. If any of the remaining six combinations were to result, an architectural violation would exist, which would be undetected by the method according to FIG. 3.

[Present application , paragraph 0019] In an additional aspect of the method the resource includes a first resource and a second resource. The set of non-unique values is a set of value-lists, and each member of the set of value-lists has a first value and a second value, the *first value being a permissible value* of the first adjacent resource, and the *second value being a permissible values* of the second adjacent resource. *Verification of a valid member* of the set is established by verifying an equality between a content of the first adjacent resource and the first value of a member; and verifying an

equality between a content of the second adjacent resource and the second value of the member.

The first passage above shows that a value or set of values can be “valid” but not be “permissible”; and the second passage above shows that a value can be “permissible” but not be “valid” — for a member of a set of “permissible values” to be considered “valid”, additional tests are necessary. Thus, the present specification is clear that being “permissible” and being “valid” are two distinct concepts, and are *not* equivalent.

In contrast, Saha, equates “valid” with “possible” (as in “*Valid or possible* set of values...”, Saha 2.2, first column on page 36, emphasis added). The present specification, however, makes numerous references to “permissible” values, and in all cases the use of the term “permissible” is distinct from the use of the term “valid” or “possible”.

Therefore, the Applicant respectfully submits that Saha’s “set of valid or possible set of values” is *not* equivalent to the *permissible* values of the present invention, and therefore Saha fails to disclose or reasonably suggest the “non-unique values” of the present invention, which are described in the present specification as sets of *permissible values*, and which are claimed in independent claims 27, 42, and 49.

The Applicant therefore respectfully submits that independent claims 27, 42, and 49 are patentable over the art of record. The remainder of the claims each depend directly or ultimately from the independent claims, and therefore are also allowable over the art of record.

In view of the foregoing remarks, all of the claims are believed to be in condition for allowance. Favorable reconsideration and allowance of the application is respectfully requested.

Please charge any fees associated with this response to Deposit Account 09-0468.

Respectfully submitted,

By:   /Suzanne Erez/                  

Suzanne Erez

Reg. No. 46,688

Phone No. 1-888-260-5928

Date: 12 January 2011

IBM Corporation

Intellectual Property Law Dept.

P. O. Box 218

Yorktown Heights, New York 10598